Appl. No. 09/871,458
Reply to Final Office Action of April 20, 2005

REMARKS

Claim Rejection Under 35 U.S.C. § 103

In the Final Office Action, the Examiner rejected claims 1, 2, 14, 15, 27, 28 and 39-42 under 35 U.S.C. 103(a) as being unpatentable over Ozzie et al. (USPN 6,640,241) in view of Rosenthal (USPN 5,964,844). Applicant respectfully traverses the rejection. The applied references fail to disclose or suggest the invention defined by Applicant's claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

Claims 1 and 14

With reference to independent claims 1 and 14, the Examiner continues to overlook certain elements of Applicant's claims. Applicant's claims 1 and 14 require receiving a command line interface (CLI) command and, in response to the CLI command, accepting commands encoded in accordance with an extensible markup language. Thus, Applicant's claims 1 and 14 specifically require a unique command line interface that is able to accept encoded commands in response to receiving a specific CLI command. None of the reference, either singularly or in combination, teach or suggest these features.

In rejecting Applicant's claims, the Examiner relied on Ozzie et al. (Ozzie) in view of Rosenthal. In general, Ozzie describes a technique that allows users at various remote sites to share and edit documents on a peer-to-peer basis. FIG. 3, relied upon by the Examiner, is a block diagram of an Internet-based system, showing both a client-server system for the WWW and a peer-to-peer system for a personal Web.

The Examiner asserted that Ozzie discloses "in response to [a] command, accepting commands encoded in accordance with an extensible markup language (using XML in processing requests)" and cited col. 11, line 22 to col. 12 line 54. The Examiner's logic is incorrect for at least two reasons.

First, Ozzie does not describe accepting commands that are encoded in accordance with an extensible markup language. In col. 11, Ozzie states that an identity manager 410 maintains a record or table, preferably in XML, of the identities of participants or subscribers and their

-2-

Appl. No. 09/871,458
Reply to Final Office Action of April 20, 2005

devices' URLs. In col. 12, Ozzie states each document being stored may internally use XML

tags. In col. 14, Ozzie states that a document's "delta" (defined as "a self-contained unit of data

that contains one or more tool-to-engine data change requests" for the document) may <u>contain</u>

commands. However, Ozzie does not describe the API itself as being capable of receiving

encoded commands.

Thus, with respect to XML, it appears the Ozzie merely describes document management

techniques in which documents or tables may contain XML tags for processing by an engine.

Thus, the Examiner is incorrect in concluding that Ozzie teaches receiving from the client a

command; and in response to the command, accepting commands encoded in accordance with an

extensible markup language. Ozzie does not teach or suggest accepting encoded commands at

all. To the contrary, the documents stored and retrieved by the Ozzie system in the web

environment contain XML, and certain tables may be stored in XML format, but Ozzie does not

describe a system in which the commands themselves received from a client are encoded in

accordance with an extensible markup language. With respect to claim 1, the Examiner has

identified no subject matter to support his conclusion that the Ozzie system utilizes encoded

commands.

Second, the Examiner is incorrect in asserting that Ozzie teaches receiving a command

and then, <u>in response to the command</u>, accepting encoded commands, as specifically required by

the Applicant's claims 1 and 14. Ozzie provides no such teaching or suggestion of a command

that, when received, causes the acceptance of encoded commands. In reference to these

requirements, the Examiner merely stated that Ozzie uses XML when processing requests from

clients. Although the Examiner is correct that the documents and tables in the Ozzie system may

have internal XML tags, this fact fails to address Applicant's requirements of receiving a

command and then, <u>in response to the command</u>, accepting encoded commands. The Examiner

fails to point to any portion of Ozzie that describes a command that results in acceptance of

subsequent encoded commands. As discussed above, the portion of Ozzie relied upon by the

Examiner merely describes a framework by which a peer-to-peer network uses XML tags within

documents or tables. Thus, the Examiner's statement that Ozzie discloses "in response to [a]

command, accepting commands encoded in accordance with an extensible markup language

(using XML in processing requests)" is incorrect.

-3-

Appl. No. 09/871,458
Reply to Final Office Action of April 20, 2005

For at least these reasons, the Examiner's assertions with respect to the primary reference, Ozzie, are incorrect. Nevertheless, the Examiner seeks to overcome the deficiencies of Ozzie based on the teachings of Rosenthal. The Examiner asserted that it would have been obvious to one of ordinary skill to modify the Ozzie system to implement a command line interface (CLI) in view of Rosenthal, and then further provide a command that, when received, results in the acceptance of encoded commands via the CLI interface.

Rosenthal describes a machine vision system. The portion of Rosenthal relied upon by the Examiner describes a vision processor capable of receiving text commands via a CLI. Like Ozzie, Rosenthal fails to teach or suggest accepting commands that are encoded in accordance with an extensible markup language. Further, like Ozzie, Rosenthal fails to teach or suggest receiving a command and then, in response to the command, accepting encoded commands. In fact, Rosenthal adds nothing other than the fact that command line interfaces (CLI) are well known.

For at least these reasons, Ozzie in view of Rosenthal fails to teach or suggest receiving from a client a CLI command and, in response to the CLI command, accepting commands encoded in accordance with an extensible markup language.

Further, Applicant respectfully points out that, contrary to the assertions of the Examiner, "outbound delta store 318" is not a router. Ozzie states that outbound deltas are stored as necessary in an outbound delta store 318 (FIG. 3). Thus, outbound delta store 318 is a storage area for outbound deltas, i.e., document change requests.

In response to Applicant's previous arguments with respect to claims 1 and 14, the Examiner merely stated that Applicant is attacking the references individually and not in combination, and cited In re Keller (1981) and In re Merck (1986). Applicant submits that the Examiner did not appreciate Applicant's response. It is well known that to establish a prima facie case of obviousness, the prior art must teach or suggest all of the claim limitations.[1] Where none of the cited references teach or suggest a limitation, the rejection must be withdrawn. Applicant has demonstrated that none of the references teach or suggest receiving a command and then, in response to the command, accepting encoded commands. Even if the Ozzie peer-to-peer system were modified in view of Rosenthal, the Ozzie system would not achieve a system

---

[1] See, e.g., MPEP 2142 citing In re Caeck, 947 F.2Dd 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

-4-

that receives a command and, in response to the command, receives commands encoded in accordance with an extensible markup language. Thus, none of the references, either singularly or in combination, teach or suggest these elements of Applicant's claims 1 and 14.

Claims 2, 15 and 28

With respect to claims 2 and 15, neither Ozzie nor Rosenthal, either singularly or in combination, describes replacing the CLI process with a management server process in response to the CLI command, wherein the management server process provides an extensible markup language-based application programming interface (API) to the client.

In rejecting claims 2 and 15, the Examiner relied almost exclusively on the Ozzie reference, stating that Ozzie describes "replacing a process with a management server process that provides an extensible markup language-based API to the client." For support, the Examiner stated that Ozzie describes processing data using APIs and cited col. 12, ll. 12-53 and col. 13, ll. 9-67.

However, these portions of Ozzie merely state that the Ozzie system "also implements application programming interfaces (API) for interacting with background services." Thus, Ozzie supports both a user interface and separate APIs, which is a well known approach. Ozzie makes no mention of "replacing a process" as asserted by the Examiner. Further, Ozzie does not describe the APIs as XML-based, as characterized by the Examiner. To the contrary, the documents managed by the Ozzie system may include XML-tags, and the API may be used to create commands for filling the delta, but there is no indication that the APIs themselves accept XML-based commands. In any event, Ozzie fails to teach or suggest replacing an interface in response to a command, as suggested by the Examiner. Again, Rosenthal fails to overcome these deficiencies. For at least these reasons, Ozzie in view of Rosenthal fails to teach or suggest replacing a CLI process with a management server process in response to a CLI command, as required by Applicant's claims 2 and 15. The Examiner's analysis with respect to claims 2 and 15 is clearly erroneous, and the rejection should be withdrawn.

Appl. No. 09/871,458
Reply to Final Office Action of April 20, 2005

Claim 27

With respect to claim 27, for at least the reasons set forth above, the applied references, either singularly or in combination, fail to teach or suggest a management server module that receives CLI commands from the CLI module and, in response to one of the CLI commands, accepts commands encoded in accordance with an extensible markup language.

Further, contrary to the Examiner's assertion, Ozzie does not even disclose a network router. Instead of referring to "relay agent" (316), as in the last Office Action, the Examiner now asserts that "outbound delta store 318" is a router. However, Ozzie states that relay 316 will receive such outbound deltas, store them as necessary in an outbound delta store 318 (FIG. 3), and forward them upon the destination peer unit 314B-D. Thus, outbound delta store 318 is a storage area (i.e., a repository) for outbound deltas, which is specifically defined by Ozzie as "a self-contained unit of data that contains one or more tool-to-engine data change requests." In no manner can outbound delta store 318 be reasonably construed as a router.

Claims 39-42

For at least the reasons set forth above, the applied references, either singularly or in combination, fail to teach or suggest receiving a CLI command and, in response to the CLI command, providing an application programming interface (API) to receive configuration requests and operational requests encoded with extensible markup language tags.

Further, contrary to the Examiner's assertion, Ozzie fails to teach or suggest accessing a network management interface schema that maps the extensible markup language tags to configuration and operational information associated with software modules running on the network router; parsing the configuration requests and the operational requests; accessing the corresponding configuration and operational information associated with the software modules according to the network management interface schema; and emitting replies encoded with extensible markup language tags according to the network management interface schema, as recited by Applicant's claim 39.

As noted above, Ozzie does not even disclose a network router. The Examiner now asserts that "outbound delta store 318" is a router. However, Ozzie states that relay 316 will receive such outbound deltas, store them as necessary in an outbound delta store 318 (FIG. 3),

-6-

Appl. No. 09/871,458
Reply to Final Office Action of April 20, 2005

and forward them upon the destination peer unit 314B-D. Thus, outbound delta store 318 is a storage area (i.e., a repository) for outbound deltas, which is specifically defined by Ozzie as "a self-contained unit of data that contains one or more tool-to-engine data change requests." In no manner can outbound delta store 318 be reasonably construed as a router.

Further, as described above, Ozzie does not describe an API that receives configuration requests and operational requests encoded with extensible markup language tags. To the contrary, the documents managed by the Ozzie system include XML-tags, but there is no indication that the APIs themselves are XML-based. Further, Ozzie fails to teach or suggest replacing and providing the API in response to a command, as suggested by the Examiner.

In addition, Ozzie fails to describe a network management interface schema that maps the extensible markup language tags from encoded requests to configuration and operational information associated with software modules running on the network router. Ozzie does not describe encoded commands, and nowhere does Ozzie describe operational information for a network router. The only mention of XML by Ozzie refers to the documents or document deltas (and not to commands received by the API itself or any other type of interface), and the API provides access to manage and edit documents for a web environment (not operational information).

Moreover, Ozzie fails to describe emitting replies encoded with extensible markup language tags. With respect to the "tags with engine identifiers" cited by the Examiner, Ozzie refers to the delta itself as being able to <u>contain</u> commands, and that APIs are provided for "filling the delta" with commands based on a user's intent.[2] However, Ozzie does not describe the API itself as being capable of receiving encoded commands or emitting encoded replies.

With respect to claim 40, neither Ozzie nor Rosenthal, either singularly or in combination, describes replacing a CLI process with a management server process in response to the CLI command, wherein the management server process provides an extensible markup language-based application programming interface (API) to the client.

Claims 41-42 are patentable for similar reasons.

---

[2] Col. 14, ll.7-20.

-7-

Appl. No. 09/871,458
Reply to Final Office Action of April 20, 2005

For at least these reasons, the Examiner has failed to establish a prima facie case for non-patentability of Applicant's claims 1, 2, 14, 15, 27, 28, 39-41 under 35 U.S.C. 103(a). Withdrawal of this rejection is requested.

Allowable Subject Matter

In the Office Action, the Examiner indicated that claims 3-13, 16-26 and 29-38 include subject matter not found within the prior art, and allowable if rewritten in independent form. The Applicant agrees with the Examiner's conclusion.

CONCLUSION

All claims in this application are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examiner is invited to telephone the below-signed attorney to discuss this application.

Date: By:

June 20, 2005   _Kent J. Sieffert_

SHUMAKER & SIEFFERT, P.A.   Name: Kent J. Sieffert
8425 Seasons Parkway, Suite 105   Reg. No.: 41,312
St. Paul, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

-8-